SCALARUP: SCALABLE DATA LOOKUP AND REPLICATION FRAMEWORK FOR UPDATED DATA

Deniz Altınbüken [a, *] and Öznur Özkasap [a]
[a, *] Department of Computer Engineering, Koç University, Istanbul, Turkey
{daltinbuken, oozkasap}@ku.edu.tr

Abstract

Scalability and data accessibility are significant issues in enhancing the power of mobile ad hoc networks (MANET). Scalable Data Lookup and Reactive Replication (SCALAR) framework for MANETs is our recent application layer solution standing out mostly by the scalability feature. It is composed of a virtual backbone construction algorithm, a scalable and low-cost data lookup protocol and a reactive replication scheme. However, SCALAR is based on the assumption that replicated data items are unchangeable, and hence updates to data items and related consistency problems are not considered. In this study, we propose an extended framework, namely ScalarUp, which offers replication service for updated data items. Issues of aperiodic data updates, ratio of read to write probabilities and data access behaviors of mobile users are considered for the replication decision. In ScalarUp, the cost function for replication decision is revised to utilize these issues.

Keywords: Data replication, data lookup, scalability, virtual backbone, connected dominating set

1. Introduction

Mobile ad hoc networks have gained substantial interest and popularity in recent years due to their immense potential in several fields of application. Example application areas include personal area networks, wireless peer-to-peer networks, military communication systems, and vehicular ad hoc networks. Lack of infrastructure, self-organization and mobility are the main features of such networks. MANETs consist of mobile nodes that can reconfigure the structure of the network continuously. The network is ad hoc since there is no fixed and known network structure that every other node forwards data. Every node can change its location, enter or leave the network. MANETs are adaptive networks and they are reconstructed according to such changes. Every node in the system may act as an end system as well as a router that forwards traffic. As the topology of the network is unstable, keeping the needed information for routing is one of the key issues in ad hoc networking.

Another significant issue in MANETs is data sharing. As the structure of the network is constructed in an ad hoc manner, disconnections of nodes may end up in partitioning of the network such that detached nodes are formed. For providing high accessibility to data, we employed replication of data among hosts that also act as routers in our primary work SCALAR [2]. In SCALAR, a dynamic virtual backbone is created in order to minimize the number of nodes involved in the data lookup process. The dynamic virtual backbone is created based on the approximation of minimum connected dominating set problem in graph theory. The data lookup protocol is used over the dynamic virtual backbone and hence low-cost data requests are realized. Then, the data lookup protocol is extended to use the data replication approach, which replicates frequently accessed data items to achieve data access efficiently.

We propose an extended framework, namely ScalarUp, which offers a replication service for updated data items. ScalarUp uses the dynamic virtual backbone construction and the data lookup protocols of SCALAR, and it contributes to replication protocol by considering updates to data items and related consistency issues Data update algorithms are added to the scenario to simulate the updating behavior during the data request and replication processes. The updates are done randomly and the write frequencies are stored for use in replication decision. In SCALAR, the replication decision was made according to the popularity of read-only data, and that was identified with the number of times the data is read. In ScalarUp, the replication decision is made by taking both the write frequency and read frequency of data into consideration.

ScalarUp is recommended as a fully distributed approach that operates in a peer-to-peer fashion. It is designed to operate efficiently in large-scale ad hoc networks where updates to data are allowed. To the best of our knowledge, a data replication framework for large scale ad hoc networks, where updates to data are taken into consideration, has not been investigated before.

The rest of this paper is organized as follows: Section 2 reviews related work. In Section 3, the ScalarUp framework is presented in connection to our previous work. In Section 4, the contributions of ScalarUp are discussed in detail. Section 5 presents an example scenario considering updated data, and Section 6 concludes our paper.

2. Related Work

A data replication mechanism is used to enhance data accessibility and to prevent data losses in case of disconnection or relocation of nodes in a MANET. It relies on the idea that data are replicated on nodes to maintain accessibility, even if the original owner node is not present. In this section, we review prior work on data replication techniques for MANETs as a solution to the data accessibility issue.

Hara's pioneering work on data replication considering data update [1] proposes three replica allocation methods, namely, SAF, DAFN and DCG. In the SAF (Static Access Frequency) method, mobile hosts keep replicas of data items considering their access frequencies to the data items. In the DAFN (Dynamic Access Frequency and Neighborhood) method, neighborhood of mobile hosts is also considered in addition to the access frequency. In the DCG (Dynamic Connectivity based Grouping) method, both access frequencies and the topology of the whole network are considered.

These methods are then extended considering data updates. The data updates are assumed to occur randomly and the replication decision is made according to the Read/Write Ratio (RWR) which is defined as the read probability of a data item over its write probability. Thus, a data is replicated with a higher probability if it is read more frequently than it is updated.

In the Extended SAF plus (E-SAF+) method only RWR values are considered during replication decision. So, during a relocation period every host first sends the write frequency of its own data item to other hosts. Later, the replication decisions are made depending on RWR. In the E-DAFN+ method after receiving the write frequencies every host replicates data taking the RWR into account. Later, for every set of connected mobile hosts the host with the lowest suffix becomes the coordinator. These coordinator hosts eliminate duplicated replicas in their neighborhoods using RWR values. In the E-DCG+ method after the write frequencies are broadcast, again groups of mobile hosts are created. But in the E-DCG+ method these groups are formed from biconnected elements. Later, the coordinators of the groups replicate data items in available memory spaces of the group, depending on the RWR values.

The work of [1] also discusses the effects of the write frequency underlining that the write frequency is inversely proportional with the data accessibility. It is also emphasized that E-DCG+ method has the highest data accessibility compared to other methods proposed. The effects of the on-schedule probability, the relocation period, the radio communication range, the skew of access and write frequencies have also been investigated.

Another method for data replication in MANETs is DREAM [3] that considers energy consumption during data access in addition to considering data access frequency. Overall, the aim is to optimize the energy consumption in the network. Thus, data items, that have higher access frequencies and that are accessed with energy efficient connections, are preferred for replication.

In CADRE (Collaborative Allocation and Deallocation of Replicas with Efficiency) [4], a fair data replication method is proposed. The fairness is achieved through the collaboration of all nodes. The replication and removal of data items are decided together by all hosts, preventing unneeded replicas. For the decision process, every data item is assigned a score value which indicates the popularity of the data item among all nodes. The popularity of a data item is determined by its service to every node.
Therefore, the nodes with higher popularity are replicated more frequently, which ensures that the data replication is done fairly, for the advantage every node.

In CLEAR (Consistency and Load Based Efficient Allocation of Replicas) [5], both the data item to be replicated and the mobile host to act as the replica are decided by optimization functions. The data items are chosen depending on their size and access frequency. The node of replication is chosen by the load, memory and energy constraints, and access frequency of a node. As a result, the most cost effective replication is chosen.

In DHTR (Distributed Hash Table Replication) [6] the network is divided into hierarchical groups. Each group has a cluster head [CH] managing the cluster. This approach lowers the message overhead in the network. In this method, also a distributed hash table is used to keep the state information, which achieves fast accesses. But DHTR proposes no technique for the replication decision.

Another method proposed for data replication is the *Cross-Layer Design for Data Accessibility in Mobile Ad Hoc Networks* [7]. In this method, data items are replicated for the use of a group. During the replication decision, the prediction about the next topology of the network is used. In [7] it is proposed that using the present location and velocity information of nodes prediction of upcoming topology is possible.

Our study is inspired by the main idea behind the E-SAF+ method for broadcasting and making the replication decision. The details about the procedure are discussed in the following sections.

3. ScalarUp

In this section, we first describe our system model. ScalarUp framework consists of three parts as the SCALAR [2]. The virtual backbone construction algorithm achieves that low message overhead is created, as traversing only the virtual backbone is enough for locating a data item. The scalable data lookup protocol also makes use of the virtual backbone. The reactive replication scheme is built over the scalable data lookup protocol and ensures lower message overhead and better data accessibility. Differently from SCALAR, in ScalarUp the replication decision parameters are access frequency, write frequency and the distance (hop count) of a node. In particular a data is more eligible for replication if it is farther from a node and has higher access frequency and lower write frequency.

3.1 System Model

The system consists of N nodes each having a unique identifier number. The set of all nodes is indicated as $M = \{M_1, M_2, \ldots, M_N\}$. Each node M_i is assumed to be the owner of a data item d_i and the set of data items is indicated as $D = \{d_1, d_2, \ldots, d_N\}$. A data item d_i can only be updated by its owner M_i. The read frequency α_i of data item d_i is the number of times d_i is accessed in a node. Similarly, the write frequency β_i of data item d_i is the number of times it is updated. The write frequency of the data item d_i is broadcast again by its owner M_i. For simplicity, it is assumed that every data item is of the same length which is less than the maximum transfer unit of the path.

3.2 Virtual Backbone Construction

In MANETs, every mobile host can change its position causing reconstruction of the network on a regular basis. This continuous infrastructure change introduces the need for message broadcasting in MANETs, which is a general message overhead source. Flooding of packets is most commonly used for developing the routing framework. But, when flooding occurs many redundant packets are exchanged causing the *broadcast storm problem* [8] and collisions of packets on the wireless channels.

In order to decrease extensive message overheads and prevent broadcast storm and collision problems, using virtual backbone structures is suggested by many scholars. Virtual backbones are used in the process of constructing and maintaining the general and routing structure of a MANET. Using a virtual backbone decreases the amount of excessive message overhead to a great extent by limiting the number of nodes that can broadcast a message.

We use the CDS construction algorithm [9] for constructing the virtual backbone of the MANET. A connected dominating set (CDS) of the unit-disk graph in the network topology is used during the construction of a virtual backbone.

3.3 Scalable Data Lookup Protocol

The scalable data lookup protocol aims to locate a data item and route it to the desired node. In MANETs, flooding is a commonly used technique for locating a data item but it causes large message overheads. In our design, the messages are broadcast only in the virtual backbone. Limiting the number of nodes decreases the message overhead and this provides an advantage for large-scale mobile networks.

In our protocol every node is marked as either an end system (dominatee) or a backbone (dominator) in the virtual backbone construction phase. An end system node can send a data request message only to one of its neighbors. A backbone node is responsible for the entire data lookup process which consists of the searching and receiving of the data.

The data lookup process is realized in two parts. Firstly, the data that is decided to be replicated is searched. If the data is requested by an end system, the end system sends a request message to the backbone node it is connected. Then the backbone node broadcasts this request to other backbone nodes. On the other hand, if the data is requested by a backbone, the request message is directly broadcast to other backbone nodes. When a backbone node receives a data request, it either forwards the data request or fulfills it. A data request can be fulfilled if the backbone node owns the requested data item or it can be found in its neighbors or neighbors of its neighbors.

After the data item is located, it should also be received by the requester node. So, as the second part of the data lookup process the data that is found is received. During the receiving process, backbone nodes do the primary work. If they receive a data request for a data item they own, they fulfill the request. If they receive a data packet that they have requested, they store it. If they receive a data that was requested by a neighboring backbone or an end system, they forward the data.

During the data receiving process, end systems may receive data items that they have requested. When they receive the data item they directly store it.

As outlined above, in SCALAR and ScalarUp data lookup is realized by the backbone nodes up to a great extent. Clearly, backbone nodes require more energy but as virtual backbone is dynamic, this energy consumption disequilibrium is solved as end systems become backbones and backbones become end systems.

3.4 Reactive Replication

In reactive replication, the replication decision about a data item is given after the data item is received. This way the need for extra control packets is eliminated. On the other hand, following this method a data item can be located on a closer node.

In ScalarUp, the replication decision is given using the read frequency, write frequency values of a data item and the hop count between the item and the requester node. These values are used as parameters of the cost function to determine the most suitable data item for replication. The cost function and the replication decision details are discussed thoroughly in the next section.

4. Replication Considering Data Updates

4.1. Updating Data

In ScalarUp, data updates are done randomly. When a data update event is triggered, the owner of the data item updates the data. For simplicity, it is assumed that updates take a constant amount of time. After an update is done, the owner also updates the write frequency value of the data item.

4.2. Keeping and Maintaining the Write Frequencies

In ScalarUp, there are two types of nodes. Depending on the connectivity information, a node becomes either a backbone (dominator) node or an end system (dominatee) node. A dominatee node is allowed to send data requests or write frequency messages only to its neighbors in the backbone. A dominator node on the other hand can realize the data lookup process. A dominator node can both search and receive a data. Also the replication decision for a data item is given by a dominator node. Therefore, every dominator node should keep the write frequencies of data items in addition to their access frequencies, and hop counts, which are used in SCALAR [2].

The write frequencies are broadcast to other nodes after every update event. Every write frequency broadcast message has a time-to-live (TTL) value to set an upper bound to the number of write frequency broadcast messages forwarded. Nodes receiving a write frequency broadcast message firstly update their write frequency tables and then forward the message to other nodes considering the TTL.

4.3. Cost Function

The cost function is used during the replication decision process. It uses previous information on accesses, updates and the distance of the data to the requester node. The cost function at node M_j for data item d_i is specified as follows:

$$Cost(\alpha_i, \beta_i, h_{ij}) = \frac{\alpha_i}{\sum_{k \in D} \alpha_k} \div \frac{\beta_i}{\sum_{k \in D} \beta_k} * h_{ij}$$

where α_i is the read frequency of data item d_i, β_i is the write frequency of data item d_i and h_{ij} is the hop count between node M_j and the node that data item d_i will be retrieved from. D is the set of all data items in the system. Using this function, data items which are more probable to be requested, less probable to be updated soon and farther from the requester node are assigned higher cost values. These cost values are used as primary parameter in the replication decision function.

4.4. Replication Decision

Replication decision given by a backbone node may take place in two forms:

- if the backbone receives the data item on its own request, then it caches the data item only if its cost is as large as the cost of the cheapest data item in the memory or if there is an available memory space.
- if the backbone receives the data item through a forwarded request, then it checks if it is the mid node between the original sender and receiver. If it is the mid node, it may decide to replicate the data item depending on the cost value.

On the other hand, as end systems cannot forward data, only the first replication decision form can be realized by the end systems. Again cost values are used to choose between data items.

4.5. Invalidating Old Replicas

Another crucial point in considering data updates is to be able to invalidate old replicas. When a data is updated, all of its replicas throughout the network become invalid. So these replicas should be marked to prevent further copying and usage.

The procedure of invalidating old replicas is straightforward. When a node updates its data item, it sends a broadcast message to its neighbor nodes. When a node receives a write frequency broadcast message, it checks if the new frequency is greater than the write frequency it is holding. If the new write frequency value is changed the node also checks its data items for the replica of the updated data. If an old replica is found, it is marked and the memory space reserved for that data item becomes available. Hence, accesses to old replicas are avoided.

5. Example Scenario

In this section, we provide an example scenario to outline the ScalarUp framework. During the discussion of the scenario, at a given time only one node's actions will be described for simplicity.

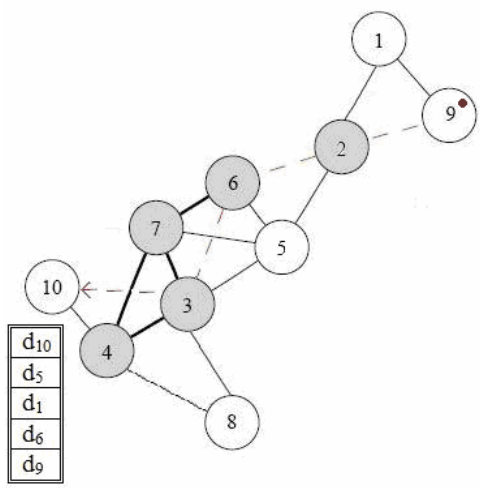

Figure 1. Example Scenario: Node9 updates data item 9.

In the beginning of the scenario, the view of the system is as shown in Figure 1. There are 10 nodes, and 5 of them construct the virtual backbone of the system. The connections between the backbone nodes are shown in bold to emphasize the paths broadcast messages may flow. In the figure, the memory space of node10 is presented. Initially node10 is holding the replicas of data items 1, 5, 6 and 9 in addition to its own data item 10. As shown in the figure, at this point data item 9 is updated and a write frequency message is *injected* from node9. The write frequency message is broadcast between backbone nodes and at last received by node10. Here it should be noted that every node in the system receives the write frequency broadcast message, but in the figure only the path from node9 to node10 is indicated with dashed lines.

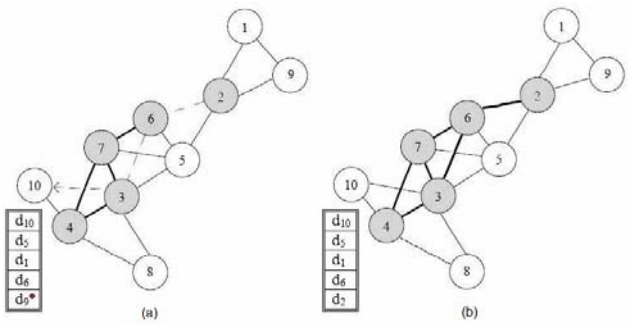

Figure 2. Example Scenario: a) The replica of data item 9 is invalidated in node10. b) Data item 2 is replicated instead of data item 9 in the next replication round.

On receipt of the write frequency broadcast message from node9, node10 updates its memory space. As data item 9 is updated, the old replica is invalidated and the memory space allocated for the old replica becomes available in the next replication round. In Figure 2(a), the invalid replica is indicated with a red mark in the memory space. In the next replication round, node10 first calculates the cost values of data items. In this example, it is assumed that data item 2 has a higher cost than data item 9 at the decision time. This can be due to the changes in the read and write frequency values. Since the last replication data item 2 may be accessed more frequently and updated less frequently, making its cost higher than data item 9's cost.

Also for simplicity, it is assumed that data items 1, 5 and 6 still have higher cost values in comparison to other data items. So node10 chooses to replicate data Item 2 instead of data item 9. The data request message and the data packet follow the path between node2 and 10, shown with dashed lines.

In Figure 2(b), the system is shown in equilibrium. At this time, there is no traffic shown between the nodes. Node10 keeps its own data item and the replicas of data items 1, 2, 5 and 6.

As illustrated with this scenario, data updates are important for the replication decisions. Data updates change the write frequency values and invalidate the replicas of the updated data item. So, every update causes reconstruction of the memory spaces in the next replication round.

6. Conclusions

We have discussed our extended SCALAR framework for updated data, namely ScalarUp. ScalarUp presents a scalable data lookup and reactive replication for updated data. ScalarUp is redesigned to support data updates, therefore it still supports the data accessibility, little message overhead and scalability features of SCALAR.

In ScalarUp, data are updated randomly by the owner of a data item. After every update the new write frequency value is broadcast to the system. When a node receives a new write frequency value for a data item, it updates its write frequencies table and invalidates old replicas of the updated item. During this updating process the message overhead is changed only in terms of the write frequency broadcast messages. But as in SCALAR, the control message overhead is low. In ScalarUp, the dynamic backbone construction algorithm used minimizes the time required to search and retrieve data for replication so even if updates invalidate old replicas the necessary replications occur with the same efficiency as in SCALAR.

As further work, we will develop the simulation model of ScalarUp and examine its performance in various application scenarios. Simulation results would let us investigate the functionality of ScalarUp in detail, providing us with the necessary information for optimizing the system.

References

[1] T. Hara and S. K. Madria, “Data replication for improving data accessibility in ad hoc networks,” *IEEE Transactions on Mobile Computing*, vol.5, no.11, pp.1515–1532, 2006.

[2] E. Atsan and O. Ozkasap, “SCALAR: Scalable Data Lookup and Replication Framework for Mobile Ad hoc Networks,” *The 28th International Conference on Distributed Computing Systems Workshops,* ICDCSW, pp.327–332, 2008

[3] P. Pabmanabhan and L. Gruenwald, “DREAM: A Data Replication Technique for Real-Time Mobile Ad-hoc Network Databases,” *Data Engineering, 2006. ICDE’06. Proceedings of the 22nd International Conference on*, pp. 134–134, 2006.

[4] A. Mondal, S. Madria, and M. Kitsuregawa, “CADRE: A Collaborative replica allocation and deallocation approach for Mobile-P2P networks,” *Database Engineering and Applications Symposium, 2006. IDEAS’06. 10th International*, pp. 21–28, 2006.

[5] ——, “CLEAR: An efficient qossip-based dynamic replication scheme for mobile p2p networks,” *Proc. of International Conference on Database and Expert Systems Applications. DEXA*, 2006.

[6] H. Yu, P. Martin, and H. Hassanein, “Cluster-based Replication for Large-scale Mobile Ad-hoc Networks,” *Wireless Networks, Communications and Mobile Computing, 2005 International Conference on*, vol. 1, 2005.

[7] K. Chen, S. Shah, and K. Nahrstedt, “Cross-Layer Design for Data Accessibility in Mobile Ad Hoc Networks,” *Wireless Personal Communications*, vol. 21, no. 1, pp. 49–76, 2002.

[8] Y. Tseng, S. Ni, Y. Chen, and J. Sheu, “The Broadcast Storm Problem in a Mobile Ad Hoc Network,” *Wireless Networks*, vol. 8, no. 2, pp. 153–167, 2002.

[9] J. Wu and H. Li, “On calculating connected dominating set for efficient routing in ad hoc wireless networks,” *Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*, pp. 7–14, 1999.